DETAILED ACTION

1. This is a notice of allowance in response to the applicant's communication filed on October 07, 2008.

2. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on October 07, 2008 has been entered.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

4. Authorization for this examiner's amendment was given in a telephone interview with the Inventor Oommen B. John; the inventors' attorneys (Hassan A. Shakir (Reg. No.: 53, 922) and the Canadian counsel on November 03, 2008.

The claims have been amended as follows:

Listing of Claims:

1. (Currently Amended) A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree, which Oommen-Rueda tree has a root, branches and leaves, comprising the steps of:

(a) receiving a first character of plaintext;

(b) traversing the Oommen-Rueda tree between the root of the Oommen-Rueda tree, and a leaf of the Oommen-Rueda tree corresponding to the first character of plaintext, and recording an Assignment Value of at least one branch of the Oommen-Rueda tree so traversed;

(c) calculating a character of ciphertext related to the Assignment Value of the at least one branch of the Oommen-Rueda tree so traversed;

(d) calculating an Assignment Value for at least one other branch of the Oommen-Rueda tree related to a distribution of the ciphertext previously calculated;

(e) receiving a next character of plaintext;

(f) traversing the Oommen-Rueda tree between the root of the Oommen-Rueda tree and a further leaf of the Oommen-Rueda tree corresponding to a next character of plaintext;

(g) calculating a further Assignment Value for at least one further traversed branch of the Oommen-Rueda tree related to a further distribution of the further ciphertext previously calculated;

(h) calculating a further character of ciphertext relating to the further Assignment Value for the at least one further traversed branch of the Oommen-Rueda tree;

(i) repeating steps (e) through (h) until all of the plaintext has been processed; ~~and~~

(j) outputting the ciphertext; and

wherein the Assignment Value for at least one branch traversed is determined in accordance with a Branch Assignment Rule.

2. Cancelled.

3. Cancelled.

4. Cancelled.

5. Cancelled.

6. Cancelled.

7. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when a member of the ciphertext alphabet is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf so that that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

8. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when a member of the ciphertext alphabet is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet more frequently than other members of the

ciphertext alphabet to the branches being traversed between the root and the leaf so that that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

9. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a zero to the majority of branches being traversed between the root and the leaf when zero is under-represented in the ciphertext generated thus far, and assigns a one to the majority of branches being traversed between the root and the leaf when one is under-represented in the ciphertext generated thus far.

10. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein, when the conditional frequency of one member of the ciphertext alphabet given a particular sequence of members of the ciphertext alphabet in the ciphertext generated thus far, is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf so that the said conditional frequency of that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

11. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein, when the conditional frequency of one member of the ciphertext alphabet given a particular sequence of members of the ciphertext

alphabet in the ciphertext generated thus far, is under-represented in the ciphertext generated thus far, the Branch Assignment Rule assigns that member of the ciphertext alphabet more frequently than other members of the ciphertext alphabet to the branches being traversed between the root and the leaf so that the said conditional frequency of that member of the ciphertext alphabet is no longer as under-represented as before the assignment.

12. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet to at least one of the branches being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the frequency of at least one member of the ciphertext alphabet in the ciphertext generated thus far, with a number associated with the output of a pseudo-random number generator.

13. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary alphabet to at least one of the branches being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the frequency of a member of the binary alphabet in the ciphertext generated thus far, with a number associated with the output of a pseudo-random number generator.

14. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by a number associated with the output of a pseudo-random number generator.

15. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by comparing a number associated with the a pseudo-random number with a range equal to half the domain of the generator generating the pseudo-random number.

16. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein the Branch Assignment Rule assigns a member of the ciphertext alphabet of cardinality R to at least one branch being traversed between the root and the leaf, such assignment being determined by invoking at least two times (R minus 1) pseudo-random numbers, the domains of at least one of the pseudo-random numbers being related to the frequencies of the occurrences of the ciphertext characters generated thus far, and the domain of at least one of the other of the pseudo-random numbers having a mean of i/R for the ith branch of each node encountered in the traversal, where i is the relative position of the branch quantified by a pre-specified ordering of the branches, and the Branch Assignment Rule

being such that where the ciphertext character associated with the ith branch in the said ordering is under-represented in the ciphertext generated thus far, it is no longer as under-represented.

17. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary; the Branch Assignment Rule assigns a member of the binary ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by invoking at least two pseudo-random numbers, the domain of the first of these pseudo-random numbers being related to the frequency of the occurrence of zero in the ciphertext, and the domain of a second of these pseudo-random numbers having a mean of 0.5, and the Branch Assignment Rule being such that when any character of the ciphertext alphabet is under-represented in the ciphertext generated thus far, it is no longer as under-represented.

18. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary ciphertext alphabet to at least one branch being traversed between the root and the leaf, such assignment being determined by comparing at least the output of two invoked pseudo-random numbers, the first of which has a domain having a mean between a number associated with the frequency of zeros and the quantity 0.5, and the second of which is a pseudo-random number having a domain whose mean is 0.5, and the Branch Assignment Rule being such that where any member of the ciphertext alphabet is

under-represented in the binary ciphertext generated thus far, it is no longer as under-represented.

19. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim [[6]] 1 wherein when the ciphertext alphabet is binary, the Branch Assignment Rule assigns a member of the binary alphabet to at least one branch being traversed between the root and the leaf by utilizing at least two pseudo-random numbers, zero being assigned when a first pseudo-random number is less than a second pseudo-random number, where the generation of the second pseudo-random number is bounded between a number associated with the frequency of zeros in the ciphertext generated thus far and the quantity of one minus the said number associated with the frequency of zeros in the ciphertext generated thus far.

20. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according claim[[s]] 1 ~~or 6~~ comprising the further steps of, after at least one traversal of the Oommen-Rueda Tree, recalculating a number associated with the frequency weight of at least one of the nodes of the Oommen-Rueda tree including the internal nodes and the leaves depending therefrom, and thereafter restructuring the Oommen-Rueda Tree in accordance with a Tree Restructuring Rule.

21. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claims 12, 13, 14, 15, 16, 17, 18, or 19, comprising the further

step of receiving first key data associated with an initial seed for at least one of the generators of the pseudo-random numbers utilized by the Branch Assignment Rule.

22. (Currently Amended) [[A]] The A method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, comprising the further step of receiving second key data associated with the structure and labeling of the Oommen-Rueda Tree.

23. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, wherein the plaintext is modified prior to processing by the addition of a pre- specified prefix data stream.

24. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, wherein at least one of the steps is preformed by a suitably programmed processor.

25. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, wherein at least one of the steps is a process executed in software.

26. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, wherein at least one of the steps is a process executed in firmware.

27. (Previously Presented) A method for decoding ciphertext encoded utilizing an Oommen-Rueda Tree and utilizing a Branch Assignment Rule to provide Assignment Values for the branches depending from the root of the Oommen-Rueda Tree, comprising the steps of:

(a) receiving a first character of ciphertext;

(b) utilizing an Oommen-Rueda Tree having a structure corresponding to the Oommen-Rueda Tree initially utilized to encode the ciphertext and utilizing the same Branch Assignment Rule as utilized to encode the ciphertext to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed;

(c) if a leaf has not been reached, utilizing the same Branch Assignment Rule as utilized to encode the ciphertext to provide Assignment Values for the branches depending from the node that has been reached, receiving the next character of ciphertext, and continuing to traverse the Oommen-Rueda Tree from the node that has been reached towards a leaf, the current symbol of ciphertext determining the branch to then be traversed;

(d) when a leaf is reached, recording the plaintext character associated with the label of the leaf, the root becoming the node that has been reached for the purpose of further processing;

(e) repeating steps c and d until all symbols of ciphertext have been processed.

28. (Previously Presented) A method for decoding ciphertext, comprising the steps of:

(a) creating an Oommen-Rueda Tree structure corresponding to the Oommen-Rueda Tree initially utilized by the Encoder;

(b) receiving a first character of ciphertext;

(c) utilizing the Oommen-Rueda Tree structure, and utilizing the same Branch Assignment Rule as utilized by the Encoder to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed;

(d) if a leaf has not been reached, utilizing the same Branch Assignment Rule as utilized by the Encoder to provide Assignment Values for the branches depending from the node that has been reached, receiving the next character of ciphertext, and continuing to traverse the Oommen-Rueda Tree from the node that has been reached towards a leaf, the current symbol of ciphertext determining the branch to then be traversed;

(e) when a leaf is reached, recording the plaintext character associated with the label of the leaf, the root becoming the node that has been reached for the purpose of further processing;

(f) repeating steps d and e until all symbols of ciphertext have been processed.

29. Cancelled.

30. Cancelled.

31. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~, comprising the further step of receiving first key data associated with the initial seed for at least one of the generators of the pseudo-random numbers utilized by the Branch Assignment Rule, where first key data is associated with the first key data utilized by the Encoder.

32. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~, comprising the further step of receiving second key data associated with the structure and labeling of the Oommen-Rueda Tree, where second key data is associated with the second key data utilized by the Encoder.

33. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~ wherein when the plaintext has been modified prior to processing by the addition of a pre-specified prefix data stream, the ciphertext is modified prior to processing by the addition of an encoded pre-specified data stream in a prefix manner, and where this encoded pre-specified data stream is related to the pre-specified data stream utilized by the Encoder.

34. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~, wherein at least one of the steps is preformed by a suitably programmed processor.

35. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~, wherein at least one of the steps is a process executed in software.

36. (Currently Amended) [[A]] The method for decoding ciphertext according to claim[[s]] 27 ~~or 28~~, wherein at least one of the steps is a process executed in firmware.

37. Cancelled.

38. Cancelled.

39. Cancelled.

40. Cancelled.

41. Cancelled.

42. Cancelled.

43. (Currently Amended) [[A]] The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim[[s]] 1 ~~or 6~~, further comprising the step of subsequently encrypting the ciphertext generated by any standard encryption process.

44. (Currently Amended) [[A]] The method of decoding ciphertext according to claim[[s]] 27 ~~or 28~~, comprising the further steps of initially decrypting the ciphertext data stream using the standard decryption process associated with the encryption process, and thereafter decoding.

45. Cancelled.

46. Cancelled.

47. Cancelled.

48. (previously presented) The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 1, wherein the ciphertext possesses Statistical Perfect Secrecy.

49. (previously presented) The method for encoding plaintext to create ciphertext using an Oommen-Rueda tree according to claim 1, wherein the ciphertext output is at least one of displayed, transmitted, and stored.

Reason for allowance

5. After consideration of the applicant's remark filed on November 03, 2008; a further search and through examination of the present application, claims 1, 7-28, 31-36, 43-44 and 48-49 are found to be in condition for allowance over prior arts of record with the above authorized examiner's amendments.

6. The following is an examiner's statement of reasons for allowance:

Claim 1 includes the following features of a method for encoding plaintext to create ciphertext using an Oommen-Rueda tree which are not taught or further suggested and would not have been obvious over prior arts of record and these features are: traversing the Oommen-Rueda tree between the root of the Oommen-Rueda tree, and a leaf of the Oommen-Rueda tree corresponding to the first character of plaintext, and recording an Assignment Value of at least

one branch of the Oommen-Rueda tree so traversed; and calculating a character of ciphertext related to the Assignment Value of the at least one branch of the Oommen-Rueda tree so traversed; until all of the plaintext has been processed recursively to output the ciphertext; where the Assignment Value for at least one branch traversed is determined in accordance with a Branch Assignment Rule according to the explicit definition in applicant's disclosure of page 19 line 6 to page 20 line 1; and the Oommen-Rueda Tree explicit definition in the applicant's disclosure of page 16 line 3 to page 17: line 13.

Claim 27 includes the following features of a method for decoding ciphertext encoded utilizing an Oommen-Rueda Tree and utilizing a Branch Assignment Rule to provide Assignment Values for the branches depending from the root of the Oommen-Rueda Tree which are not taught or further suggested and would not have been obvious over prior arts of record and these features are: utilizing an Oommen-Rueda Tree having a structure corresponding to the Oommen-Rueda Tree initially utilized to encode the ciphertext and utilizing the same Branch Assignment Rule as utilized to encode the ciphertext to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed recursively until all symbols of ciphertext have been processed according to the explicit definition in applicant's disclosure of page 19 line 6 to page 20 line 1 and the Oommen-Rueda Tree explicit definition in the applicant's disclosure of page 16 line 3 to page 17: line 13.

Claim 28 includes the following features of a method for decoding ciphertext encoded utilizing an Oommen-Rueda Tree and utilizing a Branch Assignment Rule to provide Assignment Values for the branches depending from the root of the Oommen-Rueda Tree which are not taught or further suggested and would not have been obvious over prior arts of record and these features are: creating an Oommen-Rueda Tree structure corresponding to the Oommen-Rueda Tree initially utilized by the Encoder; utilizing the Oommen-Rueda Tree structure, and utilizing the same Branch Assignment Rule as utilized by the Encoder to provide the Assignment Values for the branches depending from the root, traversing such Oommen-Rueda Tree from the root towards a leaf, the first character of ciphertext determining the branch to then be traversed recursively until all symbols of ciphertext have been processed according to the explicit definition in applicant's disclosure of page 19 line 6 to page 20 line 1 and the Oommen-Rueda Tree explicit definition in the applicant's disclosure of page 16 line 3 to page 17: line 13.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. See the notice of reference cited in form PTO-892 for additional prior art.

Contact Information

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Techane J. Gergiso whose telephone number is (571) 272-3784 and fax number is (571) 273-3784. The examiner can normally be reached on 9:00am - 6:00pm. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Emmanuel Moise can be reached on (571) 272-3865. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Techane J. Gergiso/

Examiner, Art Unit 2437

/Emmanuel L. Moise/

Supervisory Patent Examiner, Art Unit 2437